PROSILICA
Tel: 604.875.8855 *Fax*: 604.875.8856 *Web*: www.prosilica.com

GigE SDK 1.20 – Release Notes

Additions

- Registers read and write API calls added to .NET and Java wrappers.
- Added *–m* switch to sample code *CamAttr*, to get the "impact" of a particular attribute
- Existing support for ARM (LE) processor merged in the Linux SDK (for device such as the TS-7800 by Technologic Systems, http://www.embeddedarm.com).
- Adapters *bounding* is now supported on Linux.

Changes

- Attributes *ExposureValue*, *GainValue*, *WhitebalValueBlue* and *WhitebalValueRed* can now be written to when in *Auto* or *AutoOnce* mode.
- Attribute *WhitebalAutoOutliers* was removed.
- Category */Acquisition/Trigger* was added to the impact of the attribute *PixelFormat.*
- Attribute FrameRate was added to the impact of the attribute *PixelFormat.*
- Attributes *GainLevel* and *GainMode* are set to volatile when in *Auto* or *AutoOnce* mode.
- Sample code *GetXMLFile* was re-written to take advantage of the API *PvMemoryRead()* function.

Issues resolved

- Fixed frames losses when running large CCD camera (e.g GE-4900) at low frame rate.
- Fixed issue occurring on Linux when the host computer had been running for an extended period of time (Segmentation Fault upon launching an application using the API).
- Fixed issue occurring on Linux for application making active use of SIGALRM.

Notes

- **Vista User** : *PvCaptureAdjustPacketSize()* determines the maximum packet size by sending out test packets of increasing packet sizes, up to the *MaximumPacketSize* function parameter (typically set to 8228). However on some Vista network card drivers, if the driver receives a test packet of size greater than what the network card supports, the driver fails, requiring you to manually disable and re-enable the adapter. The Intel Pro 1000 GT is known to do this, whereas the Intel Pro 1000 PT is fine. Other brands or models may also do this.

GigE SDK 1.18 – Release Notes

Additions

- New supported platform
    - Mac OS X (10.4 and up) for x86, ppc and x64
- Filter Driver
    - Vista x64 support (Signed driver)
- New sample code (for all supported platforms):
    - *HardTrigger*, shows how
- New API wrapper (for Windows, Linux and Mac OS X)
    - *Java* wrapper and samples code
- New attributes (limited to Firmware 1.32 and higher, specific to cameras that support set feature):
    - *GainAutoMin*
    - *GainAutoMax*
    - *GainAutoTarget*
    - *GainAutoRate*
    - *GainAutoOuliers*
    - *GainAutoAdjustTol*
    - *GainAutoAdjustDelay*

Changes

- To the sample code *SampleViewer (in Windows)*:
    - Modified to be compatible with Visual Studio 2005
    - Moved the automatic packet size adjustment from the opening of the camera, to the start of the live view, in order to work-around a Vista issue.
- API functions *PvRegisterRead()* and *PvRegisterWrite()* performance was improved when used to read or write from multiple registers.

Issues resolved

- NDIS Filter driver was dropping packets in some particular conditions

Notes

- Java

  The Java folder contains a JNI interface to PvAPI, plus a set of samples. You will need to use the build.xml file located in each subdirectory to import the project within *Eclipse*. Each of the following samples: JListAttributes, JListCameras, JSnap, JStream, JThread, JThread3 need to have *PvJPI* in its build path. For convenience, the JNI dynamic library has been built and placed in the *bin-pc* folder. Each of the Java samples need in its Run/debug settings to have

the following added to its VM argument: -Djava.library.path=/path/to/the/SDK/bin-pc/x86. The working directory wills also have to be /path/to/the/SDK/bin-pc/x86.

- Mac OS X

  A route for 255.255.255.255 **must be** added to point to the adapter that will be plugged to the camera (or to the switch on which the camera will be). This can be done with the following command to be entered from within a Terminal:

  > sudo route -n add 255.255.255.255 169.254.42.97

  Where 169.254.42.97 is the IP (self-assigned or assigned by you) of the adapter on which the camera (or the switch) is plugged.

GigE SDK 1.16 – Release Notes

Additions

- New sample code (for all supported platforms):
  - *CamMemory*, read/write to camera user memory (Firmware 1.28 and up)
  - *CamSetup*, facilitates saving camera parameter settings to a text file as well as loading camera parameters from a text file to the camera
  - *Threading3*, similar to the sample *Threading2 (enque 3 frames rather than 1)*
- New sample code (for Windows only):
  - *Snap* for .NET
  - *Visual Basic 6.0* wrapper and samples
- New attributes (limited to Firmware 1.28 and higher, specific to cameras that support set feature):
  - *DefectMaskColumnEnable*
  - *DefectMaskPixelEnable*
  - *SyncInLevels*
  - *ModelName*
- New API functions:
  - *PvMemoryRead* and *PvMemoryWrite* (see *PvRegIo.h*)
  - *PvCaptureAdjustPacketSize*
- New Pixel formats:
  - *Bgr24*
  - *Rgba32*
  - *Bgra32*
- New error code (Windows platform only):
  - *ePvErrFirewall*, returned when Windows Firewall is blocking the streaming port

Changes

- To the sample code *SampleViewer*:
  - Added exporting of camera parameters to a text file
  - Added call to *PvCaptureAdjustPacketSize* when opening a camera in master mode.
  - Added 'force mono8' feature for rendering monochrome images on the host while streaming bayer8 from the camera
  - Supports saving of a frame to disk as Windows' bitmap (BMP)
  - Supports rendering of frame with *Bgr24*, *Rgba32* and *Bgra32* pixel formats.
- Sample code *ListCameras* now indicates if a listed camera is available or in use.
- Attribute *CameraName* can now be edited to change the name of the camera.

Issues resolved

- API function *PvUtilityColorInterpolate* was missing from .NET wrapper
- NDIS Filter driver didn't support stream multicasting to several applications running on the same host computer.

GigE SDK 1.14 – Release Notes

Additions

- Additional platforms now includes:
  - Windows XP 64bit
  - Windows Vista 32bit
  - Linux 64bit (x64 only)
- New samples code (for all supported platforms):
  - *DumpCamera,* writes to a text file the value of all the attributes of a given camera.
  - *GetXMLFile*, retrieves the on-board XML file describing the camera's registers.
  - *ResetCamera*, demonstrates how to use the low level camera's registers access functions, by commanding the targeted camera to reset it-self.
  - *StreamPnp,* demonstrates the use of the plug/unplug callback to stream from a camera recently detected.
  - *Threading2*, similar to the sample Threading but using the API function PvCaptureWaitForFrameDone() and a single frame for the streaming mechanism.
  - *MultiStream,* demonstrates the simultaneous opening and streaming from any number of camera using a separate thread for each camera.
- New samples code (for Linux & QNX platform only):
  - *AAViewer*, a simple viewer using the ASCII ART library (see http://aa-project.sourceforge.net/) for rendering.
- Multicasting of the stream from the camera is now possible on all supported platforms. The following attributes were added to support it:
  - *MulticastEnable*
  - *MulticastIPAddress*
- New attributes (Firmware 1.26 and higher and when supported by the camera):
  - *AcquisitionAbort*
  - *AcquisitionFrameCount*
  - *AcqEndTriggerEvent, AcqEndTriggerMode, AcqEndTriggerDelay, AcqEndTriggerSoftware*
  - *AcqRecTriggerEvent, AcqRecTriggerMode, AcqRecTriggerDelay, AcqRecTriggerSoftware*
  - *BandwidthCtrlMode*
  - *DSPContinuousMode* and *DSPContinuousLatency*
  - *IrisMode, IrisAutoTarget, IrisVideoLevelMin, IrisVideoLevelMax* and *IrisVideoLevel*
  - *RecorderPreEventCount*
  - *StreamHoldCapacity*
- New attributes (any Firmware versions):
  - *StatFilterVersion* (only on Windows platforms)
  - *HeartBeatInterval*

Changes

- Changing the *PacketSize* attribute after a call to PvCaptureStart() will now return *ePvErrBadSequence*.
- The API call *PvAttrRangeEnum()* will return *ePvErrBadParameter* if the supplied buffer is to small to accommodate the whole range.
- Sample code SampleViewer wills now opens the camera in *monitor mode*, if the camera is already open by someone else in *master mode*.
- The sample code *ListCameras* now displays the IP address of each detected cameras.
- The attributes category *DSPSubregion* was renamed *DSP.*
- All the *Stats* attributes can now be read even if the camera was unplugged.
- Acquisition modes expanded (Firmware 1.26 and higher) : *SingleFrame*, *MultiFrame*, *Recorder*.
- The attribute *StreamHold* was renamed *StreamHoldEnable*.
- The attribute *PacketInterval* was removed.
- The value *Armed* in attributes *SyncOutXMode* and *StrobeXMode* was renamed *AcquisitionTriggerReady* and the mode *Acquiring* was added.

Issues resolved

- Range of the ROI related attribute (e.g *Width*) was not updated when a different configuration file was loaded after a change to the camera's binning setup.
- Repeated calls to *PvCaptureWaitForFrameDone()* could lead to this function call never returning.
- Calling PvCaptureQueueClear() from a separate thread could lead to that call hanging.
- Saving Bayer16 image to TIFF in SampleViewer (Windows) was saving some garbage in the file.

GigE SDK 1.12 – Release Notes

Additions

- Added dialog to the SampleViewer to seek a camera by using its IP address (needed when connecting a camera through a gateway).
- Several new attributes have been added (Applies to cameras with firmware version 1.24 and higher):
    - *MirrorX*
    - *AcquisitionFrameCount*
    - *OffsetMode* & *OffsetValue*
- New enumerate values added to the attributes *SyncOut1Mode* and *SyncOut2Mode* when supported by the camera: *SyncIn1*, *SyncIn2*, *SyncIn3* and *SyncIn4*.
- New enumerate values added to the Strobe Mode attributes (e.g. *Strobe1Mode*) when supported by the camera: *SyncIn1*, *SyncIn2*, *SyncIn3* and *SyncIn4*.
- New enumerate values added to the attribute *AcquisitionMode* when supported by the camera:
    - *SingleFrame*
    - *MultiFrame*
- Added new sample code *BatchTrigger* for all supported platforms.

Changes

- The API and Filter driver now support on-the-fly changes of the Imaging settings (ROI, Pixel format) granted the en-queued frame buffers are large enough to accommodate for the increased amount of data. If it is not the case, the status of the frame will be set to *ePvErrBufferTooSmall*.
- Attributes *SyncInGpiLevels* and *SyncIn1Mode* were removed
- Some enumerates values of the attributes *SyncOut1Mode* and *SyncOut2Mode* were renamed:
    - *Readout* becomes *FrameReadout*
    - *TriggerReady* becomes *FrameTriggerReady*
    - *Trigger* becomes *FrameTrigger*
- Some enumerates values of the Strobe Mode attributes (e.g. *Strobe1Mode*) were renamed:
    - *TriggerReady* becomes *FrameTriggerReady*
    - *Trigger* becomes *FrameTrigger*
    - *Readout* becomes *FrameReadout*
- Attribute *ConfigFileSave* is now marked as "not available" (instead of "invalid") when *ConfigFileIndex* is set as Factory. The API function *PvAttrIsAvailable()* shall be used to test for the availability of any attribute at any given time.
- In SampleViewer, the availability of an attribute is checked before allowing the user to edit it. An attribute value will always be read even if the attribute is not available.
- Change of the attribute *PixelFormat* value will trigger a refreshment of all the ROI attributes. Values will be corrected if they are out of range.
- Statistic attributes *ElapsedFrame*, *ElapsedPacket* and *FramesRecovered* were removed.
- All the statistic attributes were renamed to use *Stat* as prefix.

- The statistic attribute *FilterDriver* was renamed to *StatDriverType* and its range of values was changed to: *Standard*, *Filter* or *Performance*.
- Some attributes were renamed (only for firmware 1.24 and higher):
    - *FixedRate* becomes *FrameRate*
    - *SoftwareTrigger* becomes *FrameStartTriggerSoftware*
    - *ExternalTriggerDelay* becomes *FrameStartTriggerDelay*
    - *ExternalTriggerEvent* becomes *FrameStartTriggerEvent*
    - *TriggerMode* becomes *FrameStartTriggerMode*
- All DSP sub region attributes were renamed to use *DSP* as prefix.
- Some of GigE related attributes were renamed:
    - *DeviceAddress* becomes *DeviceEthAddress*
    - *HostAddress* becomes *HostEthAddress*
    - *HostIP* becomes *HostIPAddress*
    - *DeviceIP* becomes *HostIPAddress*
    - *StreamBytesPerSec* becomes *StreamBytesPerSecond*

Issues revolved

- API function *PvCameraInfoByAddr()* was returning *ePvErrBadParameter* when parameter pIpSettings was set to NULL.
- Sampleviewer on Windows was crashing when displaying the histogram with a Bayer16 based stream.
- *tPvFrame*'s field *FrameCount* was not set when using the Filter Driver on Windows.

GigE SDK 1.10 – Release Notes

Changes

- Latest GigE camera firmware (1.18 and higher) sends YUV data in the same format as DCAM camera, special handling isn't necessary any longer. *SampleViewer* code has been modified accordingly.
- *Threading* sample code was modified to display more information on the image stream.
- *ListAttributes*, *Threading* and *Stream* samples were modified to accept the IP address of the camera to be used (using *PvCameraOpenByAddr*) as command line argument.
- The following sample programs were added:
    - BatchStream
    - CamAttr (Linux & QNX only)
    - Ping
    - NET (PvAPI wrapper in C# .NET)
- The following functions were added to the API (requires firmware 1.16 or higher):
    - PvCameraInfoByAddr()
    - PvCameraListUnreachable()
    - PvCameraOpenByAddr()
    - PvCameraIpSettingsGet()
    - PvCameraIpSettingsChange()
- On camera with firmware 1.20 (or higher) the following attributes were added:
    - ConfigFileIndex
    - ConfigFileLoad
    - ConfigFilePowerUp
    - ConfigFileSave

Issues revolved

- *ePvErrDataMissing* instead of *ePvErrBufferToSmall* was returned when a supplied buffer was too small to fit a frame.
- Requesting the value of the attribute *PixelFormat* when the camera was opened in *Monitor* mode was causing a crash of the application.
- Filter driver was not used when application was run from a non-Admin account (on Windows).
- Some frames were been missed when stopping/restarting the stream (when not using the filter driver) without closing the camera
- The field *ImageSize* of the tPvFrame structure was not filled by the API when the frame was returned to the application.
- Attributes were not refreshed in the Linux *SampleViewer* when the impact of a modified attribute was "/".